iSYSTEM

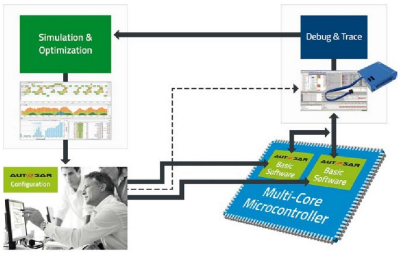
BLUEPRINT

Enabling Safer Embedded Systems

Issue #03, May 2016

Multi-core processors can seem to be the solution to a whole host of development challenges. Most application developers can assume that the underlying operating system will simply distribute the tasks to be completed across the available processing resources. Such devices prove, however, to be a double-edged sword for the automotive industry.

With the availability of single-board computers, such as the Raspberry Pi, featuring a Broadcom churning out quad-core, 1.2GHZ, 64-bit Cortex-A53 products like the BCM2837 for under $35 USD, automotive industry outsiders wonder why vehicle electronics aren't able to deliver "smartphone like" capabilities into the vehicle today. The reason is simple - safety.



A non-functional smartphone can be, of course, highly irritating, but a non-functional vehicle ECU could be, in the worst case, deadly. Vendors supplying the automotive industry have to ensure that their products meet high levels of safety, their processes are traceable and that their silicon can withstand the harsh, high-temperature environment of a vehicle. In order to guarantee such levels of quality, the resulting silicon tends to be limited in pure MIPs performance when compared to their consumer-focused counterparts. As a comparison, an Infineon AURIX™ TC23xL features a single core 200MHz processor. Even the AURIX™ TC29xTX family, with three 32-bit TriCore™ processors, have a maximum clock frequency of just 300MHz.

In order to ensure safety demands are fulfilled, the application needs to be deterministic under all conditions and capable of fulfilling its tasks regardless of the vehicle's network traffic or any algorithm's code execution it may need to complete.

In their whitepaper Elektrobit, Infineon, Timing-Architects and iSYSTEM propose a methodology and tool suites that allow a multi-core AUTOSAR based application to be analysed and then optimised to ensure that application tasks are assigned to a processing core that can ensure the entire application can meet its deterministic timing needs for applications with high ASIL-level requirements.

It is often the case that an application already exists based upon a single-core MCU. In order to reuse the available code base, whilst increasing the feature set, tools such as Timing-Architect's TA Optimizer can undertake simulations based upon system models to recommend how best to distribute tasks across the available processor cores.

The target Infineon MCU uses Elektrobit's tresos AUTOSAR OS which offers both enhanced safety capability and a core-to-core communication layer. The initial version of the application code can be executed on the target with trace data being collected by iSYSTEM's On-Chip Analyser hardware together with winIDEA. This trace data can be fed back into the TA Optimizer to improve the model, enabling further iterative improvements in task distribution.

Topics such as this are raised at the Embedded Multi-Core Conference (see 'Upcoming Events' above) and in the whitepaper listed below.

Link: bit.ly/emcc-muc-wp

Upcoming Events:

28th - 30th June 2016
EMCC, Munich

More Info @ bit.ly/emcc-muc

28th Nov - 2nd Dec 2016

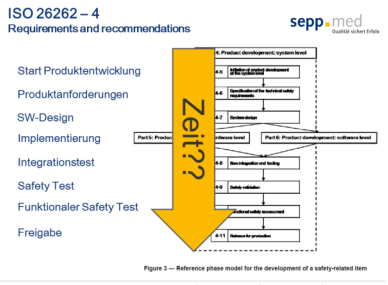
ESE Kongress, Sindelfingen

More Info @ http://bit.ly/ese-kongress-2016

When should I start with test design?

Reading a standard like the ISO 26262 for road vehicle functional safety or IEC 62304 for medical device software for software life cycle processes, it is challenging to understand 'how' to fulfil the standard. Such documents focus on clarifying 'what' needs to be achieved or fulfilled but leaves the implementation open for an organisation to decide for itself. If, however, you are confronted by such a task for the first time, you can be left feeling a little lost.

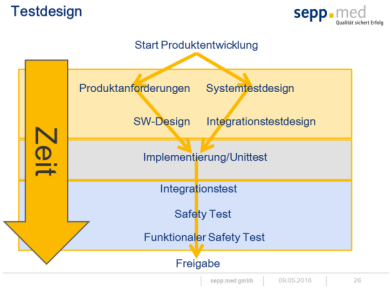
The V-Model can often be a little misleading in that, with its down phase towards unit design and implementation and up phase towards integration and testing, there is natural tendency to tag a time base along the length of the V from left to right.



There are, however, alternative lines of thought on how to interpret the approach. In our recent seminar series together with sepp.med GmbH, Dr. Beißer proposed that the product requirements and software design should run in parallel to system test design and integration test design. In addition, the unit tests should be developed parallel to the implementation of the application code (see image above-right).

"Such documents focus on clarifying the 'what' … but leave the implementation open…"

The thinking is that this approach integrates a testing strategy more closely into the development process that, in turn, avoids testing becoming a "bolt-on" activity hurriedly put together at the end of the



project. In addition, developing unit tests parallel to code development ensures that a test suite is always available to help find those little bugs that just slip in as code is "improved" or integration challenges are "fixed". Inevitably such activities are where coding errors occur as developers focus on the details to fix an error whilst the bigger picture gets forgotten for a moment.

If you were unable to join us for our seminar series, the slides from the presentation can be downloaded from the link given below.

Link: bit.ly/erst-der-test-folien

What's all this 'trace' stuff anyhow?

Our recent BugHunter blog entries have been focusing on trace, a feature that allows an engineer to see exactly what code in their application was executed at object-code level. Whether for the purposes of debugging an application or examining code coverage in order to prove the level of testing achieved, trace can ultimately deliver many insights into embedded systems that are near impossible to achieve using other methods.

Unfortunately, such development capabilities are rarely covered in today's university courses and it is often left to tenured engineers to train newer employees on such debugging capabilities. Luckily, with the prevalence of cheap, Linux-based development platforms, such as BeagleBone , it is possible to trial trace, profiling and code coverage utilising Linux's built-in command-line development tools. Our posts show how to do this and compares the results as provided under Linux and the winIDEA IDE. If you are looking for a way to give students or new employees a way to learn the concepts of embedded debugging, it may be worth taking a look!

Link: bit.ly/bughunterblog

iSYSTEM AG für Informatiksysteme
Carl-Zeiss-Str. 1
85247 Schwabhausen
Germany

Phone: +49(8138)6971-0
Fax: +49(8138)6971-46

E-mail: info@isystem.com
Web: www.isystem.com

Commercial register of the local courts (Amtsgericht) München HRB 148751
VAT identification number: DE128231221

Board of Management:
Erol Simsek, Werner Fischer, Martin Gröstenberger

Responsible for the content § 10 Absatz 3 MDStV:
Erol Simsek